在C语言中，要求运算数必须是整型的运算符是()  
A:/  
B:++  
C:!=  
D:%

若希望当A的值为奇数时，表达式的值为"真"，A的值为偶数时，表达式的值为"假"。则以下不能满足要求的表达式为( )  
A:A%2 == 1  
B:! (A%2==0)  
C:! (A%2)  
D:A%2

以下程序的运行结果是( )。  
int main( void ){  
int m = 5;  
if ( m++ > 5) printf("%d/n", m);  
else printf("%d/n", --m);  
}  
A:4  
B:5  
C:6  
D:7

以下对一维整型数组a的正确说明是( )  
A:int a(10)  
B:int n = 10, a[n]  
C:int n  
scanf("%d", &n)  
int a[n]  
D:#define SIZE 10  
int a[SIZE]

下面程序的功能是将已按升序排好序的两个字符串a和b中的字符，按升序归并到字符串c中，请为程序中的空白处选择适当的语句。 第一空答案（　 ）  
#include   
int main(void)  
{  
char a[ ] = "acegikm";  
char b[ ] = "bdfhjlnpq";  
char c[80], \*p;  
int i = 0, j= 0, k = 0;  
while( a[i] != '/0' && b[j] != ‘/0' )  
{  
if ( a[i] < b[j] ) { ( ① ) }  
else { ( ② ) }  
k++;  
}  
c[k] = ‘/0';  
if ( ③ ) p = b + j;  
else p = a + i;  
strcat ( c , p );  
puts ( c );  
}  
A:c[k] = a[i]; i++  
B:c[k] = b[j]; i++  
C:c[k] = a[i]; j++  
D:c[k] = a[j]; j++

下面程序的功能是将已按升序排好序的两个字符串a和b中的字符，按升序归并到字符串c中，请为程序中的空白处选择适当的语句。第二空答案（ 　）   
#include   
int main(void)  
{  
char a[ ] = "acegikm";  
char b[ ] = "bdfhjlnpq";  
char c[80], \*p;  
int i = 0, j= 0, k = 0;  
while( a[i] != '/0' && b[j] != ‘/0' )  
{  
if ( a[i] < b[j] ) { ( ① ) }  
else { ( ② ) }  
k++;  
}  
c[k] = ‘/0';  
if ( ③ ) p = b + j;  
else p = a + i;  
strcat ( c , p );  
puts ( c );  
}  
A:c[k] = a[i]; i++  
B:c[k] = b[j]; i++  
C:c[k] = a[i]; j++  
D:c[k] = b[j]; j++

以下程序的正确运行结果是( )。  
#include   
int f(int a);  
int main(void)  
{  
int a = 2, i;  
for( i = 0; i < 3; i++ ) printf("%4d",f(a));  
}  
int f(int a)  
{  
int b = 0;  
static int c = 3;  
b++; c++;  
return(a+b+c);  
}  
A:7 7 7  
B:7 10 13  
C:7 9 11  
D:7 8 9

以下程序的功能是：读入一行字符（如：a，b，... y，z），按输入时的逆序建立一个链接式的结点序列，即先输入的位于链表尾（如下图），然后再按输入的相反顺序输出，并释放全部结点。请在空白处为程序选择适当的代码。第二空答案（ 　）  
  
#include   
#define getnode(type) ① malloc(sizeof(type))  
int main(void)  
{  
  struct node{  
      char info;  
      struct node \*link;  
   }\*top,\*p;  
   char c;  
   top = NULL;  
   while( ( c = getchar( ) ② )  
   {  
         p = getnode ( struct node );  
         p -> info = c;  
         p -> link = top;   
         top = p;  
   }  
   while ( top )  
  {  
       ③ ；  
       top = top -> link;  
       putchar ( p -> info );  
       free ( p );  
   }  
}  
A:=='/0'  
B:!='/0'  
C:=='/n'  
D:!='/n'